

WHAT IS CLAIMED IS:

- 1 1. A method comprising:
 - 2 in a processing system, receiving a rule set as a single package;
 - 3 generating a dependency graph for the rule set; and
 - 4 generating a sequence of processing logic for optimal processing of inputted facts.
- 1 2. The method of claim 1 in which processing comprises single pass execution when there
2 are no logical loops.
- 1 3. The method of claim 1 in which processing comprises multi-pass execution when there is
2 logical loops.
- 1 4. The method of claim 3 in which processing further comprises providing an endless loop
2 terminating condition.
- 1 5. The method of claim 1 in which the rule set is free of logical conflicts.
- 1 6. The method of claim 1 in which generating the dependency graph comprises determining
2 logical dependencies across rules contained in the rule set.
- 1 7. The method of claim 6 in which generating the dependency graph further comprises
2 resolving logical conflicts using override instructions.
- 1 8. The method of claim 7 in which generating the dependency graph further comprises
2 analyzing the rule set with a business logic generation utility optimized for one of a plurality
3 of target programming languages and generating optimized logic for a selected target
4 programming language.
- 1 9. The method of claim 8 in which the target programming language is Java.
- 1 10. The method of claim 8 in which the target programming language is C++.
- 1 11. The method of claim 8 in which the target programming language is Jython.
- 1 12. The method of claim 8 in which the target programming language is JavaScript.

- 1 13. The method of claim 8 in which the target programming language is Visual Basic
- 1 14. The method of claim 8 in which the target programming language is C#.
- 1 15. The method of claim 8 in which the business logic generation utility's generated
2 processing logic comprises a series of calls to a working memory database to retrieve,
3 manipulate and update data.
- 1 16. A method for automating business processes comprising:
 - 2 in a computer system, receiving a rule set as a single package;
 - 3 determining logical conflicts within the rule set;
 - 4 resolving the logical conflicts; and
 - 5 generating a sequence of processing logic from the rule set for optimal processing of
6 inputted facts.
- 1 17. The method of claim 16 in which resolving comprises determining override conditions in
2 rule collision events.
- 1 18. The method of claim 16 in which generating comprises analyzing the rule set with a
2 business logic generation utility optimized for one of a plurality of target programming
3 languages and generating optimized business logic for the selected target programming
4 language.
- 1 19. The method of claim 18 in which the target programming language is Java.
- 1 20. The method of claim 18 in which the target programming language is C++.
- 1 21. The method of claim 18 in which the target programming language is Jython.
- 1 22. The method of claim 18 in which the target programming language is JavaScript.
- 1 23. The method of claim 18 in which the target programming language is Visual Basic.
- 1 24. The method of claim 18 in which the target programming language is C#.

- 1 25. The method of claim 18 in which the business logic generation utility's generated
2 processing logic comprises a series of calls to a working memory database to retrieve,
3 manipulate and update data.
 - 1 26. A computer program product, disposed on a computer readable medium, for business
2 processing automation, the program including instructions for causing a processor to:
3 receive a rule set as a single package;
4 generate a dependency graph for the rule set; and
5 generate a sequence of processing logic for optimal processing of inputted facts.
 - 1 27. The product of claim 26 in which the rule set is free of logical conflicts.
 - 1 28. The product of claim 26 in which to generate the dependency graph comprises
2 determining logical dependencies across rules contained in the rule set.
 - 1 29. The product of claim 28 in which to generate the dependency graph further comprises
2 instructions for causing the processor to:
3 determine logical conflicts between rules in the rule set; and
4 resolve the logical conflicts with override instructions.